GRAPHIC EDITOR

Bibliographic Fields

Publication number: JP1191270

Publication date: 1989-08-01

Inventor: MORITA YASUSHI

Applicant: HITACHI LTD; HITACHI ENG CO LTD

Classification:

- international: ***G06T11/80;*** **G06T11/80;** (IPC1-7): G06F15/62

- European:

Application number: JP19880014383 19880127

Priority number(s): JP19880014383 19880127

Abstract

PURPOSE:To attain a high speed response by retrieving graphic information by introducing a binary tree to a system for retrieving graphic information by defining a coordinate to be a key and editing and outputting. CONSTITUTION:In order to retrieve the large quantity of the graphic information, the binary tree for enclosing the graphic information, holding a rectangle coordinate (the diagonal vertex of a rectangle, for instance, coordinates of lower left and upper right vertexes of the rectangle) circumscribing thereto and constituted of plural nodes (respectively corresponding to one graphic a piece of graphic information) having a priority applied respectively by defining the coordinate to be a reference is introduced to retrieve the graphic information based on this binary tree. At the time of retrieving, all the tree nodes are not searched but the retrieval according to the priority of the node is carried out. Thereby, the target graphic information can be made access at high speed.

明細書

Specification

1.発明の名称

1.Title of Invention

図形編集装置

graphic shape editor

Claims

2.特許請求の範囲

2.Claim (s)

(1)

(1)

図形情報を入力し、記憶する手段と、前記図形情報を指定された位置に表示する手段と、前記図形情報を囲み、これに外接する矩形の１対の対角頂点の座標を各図形毎に演算し、記憶する手段と、前記各図形毎の矩形の １ 対の対角頂点の座標に基づいてバイナリ・ツリーを生成する手段とを具備したことを特徴とする図形編集装置。

Calculating co-ordinate of opposing corners apex of one pair of rectangular where itinputs graphic shape data , surrounds meaus. aforementioned graphic shape data which isindicated in location which means. aforementioned graphic shape data which isremembered is appointed. is circumscribed to this in each every graphic shape , graphic shape editor . which designates that means which forms binary *tree on thebasis of co-ordinate of opposing corners apex of one pair of rectangular every of means. aforementioned each graphic shape which you remember is possessed asfeature

(2)

(2)

前記バイナリ・ツリーは、ルート・ノードおよび他のツリー・ノードの集合よりなり、前記ルート・ノードおよび他のツリー・ノードはその下に接続される右および左の子ノードを有することを特徴と

graphic shape editor , which is stated in aforementioned Claim 1 which designatesthat it possesses child node right and left where theaforementioned binary *tree consists of gathering of root *node and other tree *node , as for

する前記特許請求の範囲第 1 項記載の図形編集装置。

(3)

前記ルート・ノードは、空ノードであるダミーノードに接続されることを特徴とする前記特許請求の範囲第 2 項記載の図形編集装置。

(4)

前記バイナリ・ツリーの各ノードは、あるノード n に対応する矩形の 1 対の対角頂点座標を k1(n), k2(n)[ただし、k1(n)≦k2(n)]、当該ノード n に対応した区間を(α, β)とし、さらに当該ノードを a、その左ノードを b、またその右ノードを c としたとき、ノード a, b,c が次の条件を満足することを特徴とする前記特許請求の範囲第 1 ないし第 3 項のいずれかに記載の図形編集装置。

イ)

ノード a、k1(a)≦k1(b), k1(a)≦k1(c),a≦k2(a)≦β、

ロ)

ノード b、a≦k2(b)≦(α+β)/2、

ハ)

ノード c、(α+β)/2<k2(c)≦β、

(5)

前記バイナリ・ツリーは、図形情報とともに記憶されることを特徴とする前記特許請求の範囲第 1 ないし第 4 項のいずれかに記載の図形編集装置。

(6)

前記バイナリ・ツリーは、図形情報とともには記憶されず、編集操作開始時に生成されることを特徴とする前記特許請求の範囲第 1 ないし第 5 項のいずれかに記載の図形編集装置。

aforementioned root *node and other tree *node isconnected under that as feature

(3)

As for aforementioned root *node , graplic shape editor . which is stated in theaforementioned Claim 2 which designates that it is conuected to the dummy node which is a empty node as special Tetsu

(4)

When each node of aforementioned binary *tree , opposing corners apex co-ordinate of one pair of rectangular which corresponds to a certain node n k1 (n), k2 (n) {However, k1 (n) <= k2 (n) }, (;al and;be) with does segment which corresponds to this said node n,fiurthermore this said node left node of a, designating the b, and right node as c, graphic shape editor . which is stated in aforementioned Claims first or any of Claim 3 which designates that node a,b, c satisfies following coudition as feature

jp1)

node a, kl (a) <= k1 (b), k1 (a) <= k1 (c), a*k2 (a) <=;be,

jp2)

node b, a*k2 (b) <= (;al +;be) / 2,

jp3)

node c, (a+;be) / 2 <k2 (c) <=:be,

(5)

As for aforementioned binary *tree , with graphic shape data graphic shape editor . which isstated in aforementioned Claims first or any of Claim 4 whichdesignates that it is remembered as feature

(6)

graphic shape editor . which is stated in aforementioned Claims first or any of Claim 5 which designates that aforementioned binary *tree is notremembered. with graphic shape data is formed at time of compilation operation start as feature

Specification

3.発明の詳細な説明

(産業上の利用分野)

本発明は、グラフィックス・システムにおける一般的な図形編集装置に係り、特に大量の図形情報の拡大・縮小表示出力や、各図形の拡大・縮小・移動・復写・削除等を高速に行うのに好適な図形編集装置に関する。

3.Detailed Description of the Invention

(Industrial Area of Application)

As for this invention, although it relates to general graphic shape editor in graphics *system , doesenlargement & reduction indication output and enlargement& reduction & movement & returning/repeating copying & thedeletion etc of each graphic shape of graphic shape data of especially large scale in the high speed it regards preferred graphic shape

editor .

(従来の技術)

[Prior Art]

各省庁・官庁等をはじめとして一般企業においても、住宅地図等の大量の図形情報を、原図等に1つ1つ手書きによって記入修正したり、保存したりする従来の方法に代って、グラフィック・ディスプレイを用いて作画、編集を行い、プロッタやLBP(レーザー・ビーム・プリンタ)等を用いて出図する、図形編集用電子計算機システムが導入されつつある。

Regarding general industry with each ministry agency & government etcas beginning, it draws in place of conventional method which in master etc iteuters corrects graphic shape data of residential ground figure or other large scale , with one one handwriting ,retains, making use of graphic *display , compiles, release of drawing it doesmaking use of plotter and LBP (laser *beam *printer) etc, electronic computer system for graphic shape compilation is being introduced.

このようなグラフィックス・システム(図形編集方式)を実現するために、グラフィック・ディスプレイをはじめとする各ハードウェアは、図形情報が出力可能範囲(例えば、グラフィック・ディスプレイであれば、1 画面分に相当するフレーム・バッファの容量)を超えた場合には、出力図形情報をクリッピングしてその範囲内に収まるように出力する機能を有している。

In order to actualize graphics *system (graphic shape compilation system) a this way, each hardware whichbegins graphic *display , when graphic shape data exceeds output enable range (If it is a for example graphic *display , capacity of frame *buffer which is suitable to 1 screen portion),clipping doing output graphic shape data , in order to be settled inside thatrange, has had function which it outputs.

例えば、1 管(画)面に描画した図形情報を拡大して表示し、画面上で図形情報を編集するような場合、全図形情報に対して拡大座標変換を行って出力したとしても、1 管面分にクリッピングして 1 画面上に正しく(誤動作なしに)表示できるようになっている。

Expanding graphic shape data which drawing is done on for example 1 tube (Picture)aspect, assuming, that it indicated. when graphic shape data is compiled on the screen , doing enlargement coordinate conversion vis-a-vis all graphic shape data , it outputted,clipping making 1 tube surface amount, (In malfunction none) it can indicatecorrectly on 1 screen , it groans.

また、グラフィック・ディスプレイを用いて、会話形式で図形情報の移動等の編集操作を実現するために、例えば特開昭 62-57078 号公報に示すような方式を用い、マウス等で指示された座標をもとに、全図形情報を検索して編集対象の図形情報をピックする方式がある。

In addition, in order to actualize movement or other compilationoperation of graphic shape data with conversation form making use of graphic *display ,, ou thebasis of, searching all graphic shape data co-ordinate which with such as mouse the display is done making use of kind of system which is shown in the for example Japan Unexamined Patent Publication Showa 62-57078disclosure , there is a system which pick does graphic shape data of thecompilation object .

ピックされた図形情報は、ブリンク等を施してディスプレイに再表示され、各編集の可否を操作者に問い合わせる。例えば、移動の実行が操作者により指示された場合は、ブリンクされた図形を管面色(地色)等で上書き描画することによって当該図形を消去する。

graphic shape data which pick is done, administering blink etc, re-isindicated in display , inquires yes or no of each compilation to the operator . When execution of for example movement display it is done with operator .graphic shape which blink is done this said graphic shape is eliminated by factthat with such as tube surface color (background color) superscription drawing itdoes.

消去図形が管面色以外の図形色部分に重なっている場合には、前記消去によって、その部分に歯抜け部を生ずる。この歯抜け部を修復するためには、消去後に、例えば特開昭 60-173677 号公報に示されるように、消去された図形により管面色にて上書きされた図形情報を、全図形情報より検索し、前記消去図形の一部を前記歯抜け部のみに、当該部分の色で再表示する。このようにして、歯抜けとなった表示画面を修復した

When elimination graphic shape is piled to graphic shape colored part amount other thantube surface color, with aforementioned elimination, tooth comingout section is caused in portion . In order rejuvenation to do this tooth coming out section, aftereliminating, as shown in for exaunple Japan Unexamined Patent Publication Showa 60-173677disclosure , with tube surface color graphic shape data which superscription is done, is searched from all graphic shape data with the graphic shape which was eliminated, in

後、ピックされた図形を移動先に表示すれば、移動操作を完了する。

only aforementioned tooth comingout section, re-indicates portion of aforementionedelimination graphic shape with color of this said portion . this requiring, after if rejuvenation doing display screen which hadbecome tooth coming out, it indicates graphic shape which pick isdone in movement destination , it completes movement operation .

(発明が解決しようとする課題)

(Problems That Invention Seeks to Solve)

しかしながら、かかる従来の図形編集方式においては、次のような問題が発生する。

But, next kind of problem occurs regarding this conventional graphic shape compilation system .

1 点目は、図形情報を表示、出図する際に発生するもので、大量の図形情報の一部しか表示、出図の対象とはならない場合でも、全図形情報を処理した後に出力しなければならないために、表示、出力時間が非常に遅くなることである。

1 point eye must indicate graphic shape data and when release of drawing doing,being something which occurs, only portion of graphic shape data of large scale ,after treating all graphic shape data must output, even with when it does notbecome with object of indication and release of drawing, because,it means that indication and output time become slow in unusual .

2 点目は、図形情報を会話形式で編集する際に発生するもので、図形のピック、消去、移動後の修復のために、全図形情報を検索する必要があるために、検索時間が長くなって応答性が悪く、誤操作の原因ともなることである。

As for 2 points eyes, when compiling graphic shape data with conversation form , beingsomething which occurs, because of rejuvenation after pick , eliminatingand movement of graphic shape , because it is necessary to search all graphic shape data , retrieval time becoming long, responsiveness is bad, it means thatbecomes also cause of misoperation .

3 点目は、上記の各問題点の故に、応答性を確保するためには、大量の図形情報を一元的に管理したり、編集したりすることができないので、少量(適当量)の図形情報ファイルに分割して、個々に編集せざるを得ず、このため、特に分割境界にまたがるような図形情報などは、複数の図形情報ファイル等に格納されるため、該図形情報を変更する場合は、複数の図形情報として編集を複数回行うことが必要となることである。

3 points eyes, in order in reason of above-mentioned each problem , toguarantee responsiveness , manage graphic shape data of large scale monistically,because it is not possible, to compile, dividing into graphic shape data file of the small quantity (suitable amount), you must compile individually, because of this , as forkind of graphic shape data etc which extends over especially division boundary , Because it is housed in graphic shape data file etc of plural , when said graphic shape data ismodified, it means that it becomes necessary to compile as graphic shape data of plural multiple times .

4 点目も、3 点目と同様の理由により、分割した図形情報単位の表示、出力しかできないために、これらを統合して連続した表示、出力を行うことは、非常に難しい。

As many as 4 points eyes, only indication and output of graphic shape data unit which is divided due to reason which is similar to 3 points eyes,integrating these because it is not possible. thing which doesindication and output which it continues is difficult to the unusual .

このように、従来の図形編集方式では、大量の図形情報を一括して取扱おうとすれば、膨大な時間がかかり、応答性に劣るため、操作者に対しては極めて不便になる。

this way, if with conventional graphic shape compilation system , lumping together graphic shape data of large scale , it tries to handle, extended time catches, becauseit is inferior to responsiveness , vis-a-vis operator quite becomesinconvenient.

一方、応答性を確保するために、図形情報を分割して取扱えば、図形情報の編集時には極めて繁雑な作業を強いられ、視認性の良い図形構報の出力ができないという問題があった。

On one hand, in order to guarantee responsiveness , dividing graphic shape data , ifyou handle, there was a problem that it can force quite complicated jobwhen compiling graphic shape data , it cannot output graphic shape structureinformation where visual recognition is good.

本発明の目的は、このような従来方式の問題点を解決し、操作者に対する応答性と、情報の一元管理・編集及び視認性の良い図形情報の出力などが全て満足される図形編集方式を提供することにある。

objective of this invention solves problem of system a this way untilrecently, monistic management & compilation and output etc of the graphic shape data where visual recognition is good of responsiveness and data for operator all are to offer graphic shape compilation system which is satisfied.

(課題を解決するための手段)

(means in order to solve problem)

本発明では、膨大な図形情報を検索するために、図形情報を囲み、それに外接する矩形座標(矩形の対角頂点、例えば矩形の左下、右上頂点の座標)を保持し、その座標を基準として各々優先順位を付けた複数のノード(それぞれが 1 つの図形情報に対応)によって構成されたバイナリ・ツリーを導入し、このバイナリ・ツリーをもとに図形情報を検索する。

With this invention , expansion in order to search graphic shape data , graphic shape data issurrounded, rectangular co-ordinate (co-ordinate of lower left , top right apex of opposing corners apex , for example rectangular of rectangular) which is circumscribed to that is kept, the binary *tree which configuration is done is introduced with node (Each one corresponds to graphic shape data of one) of the plural which attaches each priority sequence with co-ordinate as reference , graphic shape data is searched on basis of this binary *tree .

検索に際しては、全てのツリー・ノードを探索するのではなく、ノードの優先順に従った探索を行う。優先順位を示すキーとしては、図形情報を囲む矩形座標のうち、x 座標のペア―、または y 座標のペア―を用いる。

At time of searching, it is not to search all tree *uode , searchwhich you follow priority order of node is done. Among rectangular co-ordinate which surround graphic shape data as key which shows the priority sequence , xco-ordinate [peaa], or [peaa] of yco-ordinate is used.

さらに各ノードには、図形情報の存在する区間座標が対応している。以下に、その構造を定義する。

Furthermore segment co-ordinate where graphic shape data exists corresponds to each node . Below, construction is defined.

k1(n), k2(n):

k1 (n), k2 (n):

ノード n の x 座標、または y 座標のペア―。ただし、k1(n)≦k2(n)

xco-ordinate , or yco-ordinate of node n [peaa]. However, k1 (n) <= k2 (n)

α, β :

;al and;be:

ノードに対応した区間

segment which corresponds to node

a:

a:

ノード

node

b:

b:

ノード a の左ノード

Left node of node a

C:

C:

ノード a の右ノード

Right node of node a

上記のような定義の下では、本発明のバイナリ・ツリーの各ノードは以下の条件を満足する。

As description above under definition, each node of binary *tree of the this invention satisfies condition below.

1)

1)

ノード a に対して、k1(a)≦k1(b), k1(a)≦k1(c) α ≦k2(a)≦ β

Vis-a-vis node a, k1 (a) <= k1 (b), k1 (a) <= the k1 (c);al <= k2 (a) <= the;be

2)

2)

ノード b に対して、α ≦k2(b)≦(α ÷ β)/2

Vis-a-vis node b, the;al <= k2 (b) <= (;al ÷;be) / 2

3)

ノード c に対し

$(\alpha+\beta)/2<k2(c)\leqq\beta$

上記の条件を、本ツリーの全てのノードに対して再帰的に適用することにより、バイナリ・ツリーが構成できるものである。

本構造に従って、ノードの探索を行い、検索する必要のないサブツリーをチェックすることにより、不要な検索を行なわず、目的の図形情報を高速に確定し得るものである。

(作用)

図形情報の検索に際しては、前述のバイナリ・ツリー構造を用い、不必要な情報を検索しないことにより、目的の図形情報を高速にアクセスできる。

(実施例)

以下、本発明の 1 実施例を詳細に説明する。第 1 図は、本発明による図形情報の構成を示すもので、同図(A)はバイナリ・ツリーの 1 例を示す図、周囲(B)は前記バイナリ・ツリーで表わされる具体的な図形情報群の 1 例を示す図である。第 2 図は、本発明を実施するシステム構成例である。

はじめに、本発明による会話形図形編集装置の概略構成について、第 2 図を参照して述べる。本発明の図形編集装置への図形情報の入力は、グラフィック・ディスプレイ 11 やキーボード 12 等を使用して、電子計算機 13 に格納された会話形式図形編集プログラムを起動し、種々の情報の入力は、プログラムと会話しながら、マウス 14,ライトペン 15,デジタイザ 16,キーボード12等の各種入力機器(座標入力手段)を用いて行う。20 はバッチ型入力装置である。

起動されたプログラムは、操作者の指示にしたがった図形情報(例えば、日立市大みか町 1 丁目の住宅地図)を、グラフィック・ディスプレイ 11 に表示する。

表示の際は、2 次記憶装置 17 に格納された全図形情報(日立市全住宅地図)を読み出し、指示のあった図形情報(大みか 1 丁目)を検索し、適当な座標変換(拡大・縮小及びハードウェアの座標系への変換)を施し、グラフィック・ディスプレイ 11 に出力する。

3)

In node c confronting

(;al +;be) / 2 <k2 (c) <=:be

By applying to recursive above-mentioned condition , vis-a-vis the all node of this tree , binary *tree it is sonething which configuration it ispossible.

Following to this construction , you search node , you search unnecessary by the check doing sub tree where it is not necessary to search, graphic shape data of objective it is something which it can decide in high speed .

[Working Principle]

At time of searching graphic shape data , graphic shape data of day access can bedesignated as high speed by not searching unnecessary data making use of theaforementioned binary *tree structure .

(Working Example)

Below, 1 Working Example of this invention is explained in detail. As for Figure 1 , being something which shows configuration of graphic shape data with this invention , as for same Figure (A) as for figure and periphery (B)which show 1 example of binary *tree it is a figure which shows 1 exampleof exemplary graphic shape data group which is displayed with aforementioned binary *tree . Figure 2 is system configuration example which executes this invention .

In beginning, referring to Figure 2 with this invention concerning the conceptual configuration of conversation shape graphic shape editor , you express. Input of graphic shape data to graphic shape editor of this invention , using graphic *display 11 and the keyboard 12 etc. starts conversation form graphic shape compilation program which is housed in the electronic computer 13, inputs various data , while program and conversation doing, makinguse of mouse 14, light pen 15, digitizer 16, keyboard 12or other various input instrument (co-ordinate input means). 20 is batch type input device .

program which is started, indicates graphic shape data (residential ground figure of for example Hitachi City Omika-cho 1-Chome) which you follow display of operator , in graphic *display 11.

Case of indication, all graphic shape data (Hitachi City all residential ground figures) which are housed in secondary storage 17 it searches graphic shape data (Large you see 1 -Chome) which has reading , display , administers suitable coordinate conversion (Enlargement & reduction and conversion to coordinate system of hardware), outputs to graphic *display 11.

操作者は目的の画面が表示されると、本プログラムと会話形式で、各種の入力機器を用いて指示を与えることにより、図形情報の編集(新しい住宅の追加及び、取り壊された住宅の削除等)を行う。

operator when screen of objective is indicated, with this program and conversation form , does compilation (Addition of new house and deletion etc of house which isdemolished) of graphic shape data by giving the display making use of various input instrument .

例えば、新たな線分を描画する場合には、マウス 14 を用いて、グラフィック・ディスプレイ 11 上で、所望の始点および終点をクリックすると、所望の直線が追加表示される。また、既に表示されている線分の近傍をクリックすることによって、その線分をピックし、前記線分を削除したり、あるいは、さらに他の位置をクリックすることによって、その線分を移動したりする。

When for example new linear part is done drawing , when on graphic *display 11, desired origin and the terminal click are done making use of mouse 14, desired straight lines is addedis indicated. In addition, by fact that vicinity of linear part which is alreadyindicated is done click , linear part is done pick , theaforementioned linear part is deleted, or, furthermore by fact that theother location is done click , linear part is moved.

プログラムは、各縦集操作に従って、図形をグラフィック・ディスプレイ 11 に表示したり、図形情報の更新を行なったりする。操作者より編集終了の指示があった場合、プログラムは更新された図形情報を 2 次記憶装置 17 に格納して停止する。

program , following to each vertical collection operation, indicates graphic shape in graphic *display 11, renews graphic shape data . When there is a display of compilation end from operator , housing the graphic shape data which is renewed in secondary storage 17, it stops program .

また、2 次記憶装置 17 に格納した情報の中から、操作者の指示に従って、(例えば日立市大みか 1 丁目の住宅地図)を、LBP18 やブロック 19 に選択的に出力する機能も有している。続いて、本システムの図形情報の構成について述べる。

In addition, from midst of data which is housed in secondary storage 17,following to display of operator , (for example Hitachi City large you see 1 -Chome residential ground figure), it has possessed also thefunction which selectively is outputted to LBP18 and blocks 19. Consequently, you express concerning configuration of graphic shape data of this system .

第 1 図(A)は本発明によるバイナリ・ツリー(以下、ツリーと略する)1 の構造例を示し、同図(B)はこのツリーで表わされる図形情報 2 の 1 例を図示したものである。

Figure 1 (A) binary *tree (Below, tree you abbreviate) shows structural example of 1 with this invention , the same Figure (B) is something which illustrates 1 example of graphic shape data 2 whichis displayed with this tree .

ここで、ツリーの各ノード 3 はそれぞれ、個々の図形情報 4 に対応付けられている(ノード 3 に含まれる情報については、後述する)。その対応関係は、各ノードの左(または右)上に記された丸枠付数字 5 と、個々の図形情報の左上に記された丸枠付数字 6 によって示されている。

Here, each node 3 of tree respectively corresponds to theindividual graphic shape data 4, (It mentions later concerning data which is included in node 3,). corresponding relationship is shown with round frame attaching numeral 5 which was inscribedon left (Or right) of each node and round frame attaching numeral 6 which wasinscribed to left top of individual graphic shape data .

ツリーの構造は、個々の図形情報に外接する矩形 7 の 1 対角線の左下、右上の x 座標(以下、x 1(i), x2(i):i-1, 2, 3・・・等と略する)と、各ノードに対応付けられた x 座標の区間 8 に従っている。なおここでは、説明を簡単にするため、全図形情報の持つ座標空間は、第 1 図(B)に示したように、[0,N]/{0,M]の整数座標に正規化されているものとする。ツリーのルート・ノード 9 は、区間[0, N]に右上座標 x2(I)が含まれている複数の図形情報(すなわち、第 1 図(B)に示された各図形)の中で最小の左下座標 x1(i)を持つ図形情報(第 1 図の例では、折線図形)が対応する。ルート・ノード 9 の左ノード(バイナリ・ツリーで左

However in those where j*i is included, graphic shape data (With example of Figure 1 , diamond shape graphic shape) which has minimum x1 (j) (However, j*i) corresponds, in addition as for right node (With binary *tree to right side child node which diverges),while x2 (k) (However k*i) is included in segment [N/2+1,N], graphic shape data (With example of Figure 1 , rectangular graphic shape) whichhas minimum x1 (k) (However, k*i) corresponds. 0, normalization are made integer co-ordinate of M]. As for root *node 9 of tree , graphic shape data (With example of Figure 1 , curved line graphic shape) which has minimum lower left co-ordinate x1 (i) in graphic shape data (Each graphic shape which is shown in namely, Figure 1 (B)) of plural where top

側へ分岐した子ノード)は、区間[O, N/2]に x2(j)(ただし j≠i)が含まれているものの中で、最小の x1(j)(ただし、j≠i)を持つ図形情報(第 1 図の例では、菱形図形)が対応し、また右ノード(バイナリ・ツリーで右側へ分岐した子ノード)は、区間[N/2+1, N]に x2(k)(ただし k≠i)が含まれている中で最小の x1(k)(ただし、k≠i)を持つ図形情報(第 1 図の例では、矩形図形)が対応する.

right co-ordinate x2 (l) is included in segment [0, N] corresponds. As for left node (With binary *tree to left side child node which diverges) of root *node 9, in segment [O,N/2] x2 (j) 0, N]/ xco-ordinate of lower left , top right of one pair angular line of rectangular 7 whichoutside support is made individual graphic shape data (Below, x1 (i), x2 (i):i-1.2,3*** etc you abbreviate) with, you havefollowed construction of tree , to segment 8 of xco-ordinate whichcorresponds to each node . Furthermore here, in order to make explanation simple, as for the co-ordinate space which all graphic shape data has, as shown in Figure 1 (B),

以上の説明から分るように、本ツリー構造の一般的定義はつぎのようになる。

As understood from explanation above, general definition of this tree structure is following, it groans.

(1)

(1)

あるサブツリーのルート・ノードは、区間[α , β]に x2(l)が含まれており、最小の x1(l)を持つ図形情報が対応する.

As for root *node of a certain sub tree , x2 (l) is included by the segment [;al and;be], graphic shape data which has minimum x1 (l) corresponds.

(2)

(2)

前記ルート・ノードの左ノードは、区間[α ,(α + β)/2]に X2(j)(j≠i)が含まれているものの中で、最小の x1(j)(j≠i)を持つ図形情報が対応する。

As for left node of aforementioned root *node , in those where the X2 (j) (j*i) is included in segment [;al, (;al +;be) / 2], graphic shape data which has minimum x1 (j) (j*i) corresponds.

(3)

(3)

その右ノードは、区間[(α + β)/2+1, β]に x2(k)(k≠i)が含まれているものの中で、最小の x1(k)(k≠i)を持つ図形情報が対応する。

As for right node , in those where x2 (k) (k*i) is included in segment [(;al +;be) / 2 + 1,;be], graphic shape data which has minimum x1 (k) (k*i) corresponds.

上記の定義を全てのサブツリーが満足するように、第 1 図(B)の全図形に順次繰り返し適用して構成したものが、第 1 図(A)のバイナリ・ツリーである。なお、ルート・ノードを子ノードとして持つノード 10 は、ツリー操作(例えば、第 1 図において、ルートノード 9 を削除するような操作を容易にするためのダミーノードである。ツリーの各ノードは、第 1 図(A)にも示したように、当該図形情報を保持するためのポインタと、当該図形を囲んで外接する矩形の左上、右下の座標、及び図形情報へのポインタ(例えば、メモリの先頭アドレス、データ長)、左または右の子ノードへのポインタ等を保持している。また、出力結果として、図形が描画順序に依存するような場合は、ノードを描画順に連結した双方向のポインタ等をも保持することができる。

In for example Figure 1 , it is a dummy node in order to make operation of deleting root node 9 easy. Each node of tree , as shown even in Figure 1 (A), surrounding pointer and this said graphic shape in order to keep this said graphic shape data , co-ordinate , of the left top , right bottom of rectangular which is circumscribed and pointer to graphic shape data (start address , data length of for example memory), has kept pointer etc to child node of left or right . In addition, when graphic shape depends on drawing order as output result , the node also pointer etc of two-way which coupling is made drawing order can be kept. In order for all sub tree to satisfy above-mentioned definition, the sequential repeatedly applying to all graphic shape of Figure 1 (B), those which configuration are done, are binary *tree of Figure 1 (A). Furthermore, as for node 10 which has root *node as child node ,tree operation

次に、本発明の全体的な動作を説明する前に、まず、動作の中心となるバイナリ・ツリーを用いた図形情報の検索、図形情報の追加、削除の手順につき、ツリートラバースとノード操作を重点にして説明する。なお、これらの操作手順を、第 1 図の具体的なバイナリ・ツリーに適用する例

You explain tree traverse and node operation next, before explaining the entire operation of this invention , first, concerning protocol of searching graphic shape data which uses binary *tree which becomes center of operation,adding and deleting graphic shape data , in importance. Furthermore, you explain afterwards concerning example whichapplies

については、後で説明する。

these operating sequence , to exemplary binary *tree of Figure 1 .

まず、第 4 図に示すフローチャートを参照して、バイナリ・ツリーの構造に従った図形情報の検索について述べる。

First, referring to flowchart which is shown in Figure 4 , you expressconcerning searching graphic shape data which you follow construction of the binary *tree .

グラフィック・ディスプレイに表示すべき範囲を、第 13 図に示したように、領域 30[0,N]×[0, M]上(例えば、日立市全体の地図)の[X1,Y1]x[X2, Y2]の範囲 32(例えば、大みか 1 丁目のが含まれる範囲)とする。即ち、表示しようとしている図形情報の全て、またはー部が[X1, Y1]x[X2, Y2]の範囲にあるもの(第 13 図において、符号 34,36 であらわされたもの)を、全図形情報の中から検索するものとする。

As shown range which it should indicate in graphic *display , in the Figure 13 , range 32 of [X1, Y1] x [X2,Y2] of (map of for example Hitachi City entirety) on region 30 [0, N] X [0, M](for example large you see and 1 -Chome range where is included) with it does. Namely, search thing (In Figure 13 , those which are displayed with code 34, 36.) where all , or part of graphic shape data whichit has been about to indicate is a range of [X1,Y1] x [X2, Y2], from themidst of all graphic shape data .

検索時のツリートラバース(探索)は、与えられた前記座標X1, Y1, X2, Y2を基準とし、各図形情報を囲む矩形領域の座標をこれと比較して行う。また、ツリートラバースは、当該ツリーのルート・ノードより行なう。たどったノード(現在検索中のノード)を n(401)とし、ノードに対応した区間[α , β]を[0,N]とする(402)。

As for tree traverse (Search) when searching, it designates aforementioned co-ordinate X1,Y1,X2,Y2 which is given as reference , it does co-ordinate of rectangular area which surrounds each graphic shape data by comparison with this. In addition, it does tree traverse , from root *node of this said tree . node (node which presently is in midst of searching) which it traces is done n (401) with, segment [;al and;be]which corresponds to node is done [0, N] with, (402).

つぎに、検索中のノード n が空ノードであるか、換言すれば、次にたどるべき子ノードを持っているか否かをチェックする(403)。空ノードならば、410 の判定を行ない、空ノードでなければ 404 の判定を行なう。

If next, node n which is in midst of searching is empty node or and you rephrase, whether or not which has child node whichit should trace next is done check , (403). If it is a empty node , 410 it decides, it is not a empty node , 404it decides.

ここで、検索対象ノードに対応した図形情報を囲む矩形領域の左下および右上隅の座標を、第 1 図(B)のように、{X1(n). Y1(n)}, {x2(n), y2 (n)}とし、さらに x1(n)≦x2(n), y1(n)≦y2(n)とする。

Here, lower left of rectangular area which surrounds graphic shape data whichcorresponds to retrieval object node and co-ordinate of top right corner , like Figure 1 (B). {X1 (n), Y1 (n) }, {x2 (n), y2 (n) } with it does, furthermore x1 (n) <= x2 (n), the y1 (n) <= y2 (n) with does.

また、ツリートラバースは、以下の説明から明らかなように、深さ優先で行なう。換言すれば、403 の判定が否定で、検索対象ノードに子ノードがある限りツリーを下方へたどり、たどった順にノードおよび区間をスタック(後入れ先出し)方式で記憶する。

In addition, tree traverse , as been clear from explanation below, doeswith depth priority. If you rephrase, decision of 403 being negative , if there is a child node in search object node , tree is traced to lower , in orderwhich is traced node and segment are remembered with stack (Pushdown) system .

バイナリ・ツリーの構成から明らかなように、現在の検索対象ノードの持つ矩形の左下 x 座標 x 1(n)が出力範囲の右側(例えば、第 13 図の矩形 38)であれば{X2<x1(n)}、当該ノード以下のサブツリーは、全て出力範囲外なので、検索しない(404 の前半の条件)。

As been clear from configuration of binary *tree , if lower left xco-ordinate x1 (n) of the rectangular which present search object node has is right side (rectangular 38 of for example Figure 13) of output range , {X2

また、現在のノードに対応する区間[α , β]の終点(β)が出力範囲の左側(β <X1)であれば、それ以下のサブツリーは全て左側、すなわち範囲外となるので検索しない(404 の後半の条件)。

In addition, if it is a left side (:be <X1) of terminal (;be) power output range of the segment [;al and;be] which corresponds to present node , because sub tree of less than that it becomes all left side , namely out of range , it does not

前記の判定 404 で範囲外とされたノード以外のものについては、そのノードの持つ矩形が出力範囲[X1, Y1]x[X2. Y2]に含まれるかどうかを判定する(405)。すなわち、405 の判定が不成立ならば範囲外であり、一方、同判定が成立するノードは範囲内であるので、出力の対象として保持する(406)。

続いて、現在の検索対象ノードとその区間[α , β]をスタック方式でプッシュ(入力)し(407)、[α , β]の区間計算を行なって、その左のノードにトラバースし(408,409)、判定 403 へ戻る。

このとき、もし 403 の判定が成立し、ノードが無く空ノードであると判定されれば、410 でスタックが空か否かを判定する。スタックが空でないと判定されたときは(410)、スタックをポップし(411)、右のノードにトラバースし(412,413)、判定 403 へ戻り、404 以降の判定、操作を繰り返す。判定 403 の結果が肯定でノードが無く(空ノードで)、しかも判定 410 の結果も肯定でスタックも空であるならば、全ての所望ノードについての検索が終了したことになるので、処理を停止する。

続いて、第 5 図に示すフローチャートを参照して、ツリー・ノードの生成、追加処理について述べる。n, α , β ,x,y 等は前記検索処理の場合の仮定と同様である。またノード生成動作をより容易に実現するために、ここではルート・ノードを左に持つダミーのノードより処理を開始するものとする。

新しく挿入すべきノードを new とし(501)、ツリーの構造にしたがって各ノードをたどる。すなわち、Pn をダミーノード(第 1 図の 10)とし(502)、Pn の左ノードを n にセットする(503)。in に「左」を示すフラグ値をセットし(504)、区間(α , β)に{0,N}をセットする(505)。

つぎの判定 506 では、n が空ノードか否かをチェックし、空ノードでなければ判定 507 を行なう。この判定 507 において、現在の検索対象ノードの矩形の左下 x 座標 x1(n)が、挿入すべきノードの矩形の左下 x 座標 x1(new)よりも大きければ、挿入すべき位置は、現ノードの位置であるので、現在のノードと挿入すべきノードの各ポインタを付け替えることにより、両ノードを、交換し(508,509)、現在のノードを挿入すべきノードとする(510)。

更に、挿入すべきノードの矩形の右上 x 座標 x2(new)が{ α ,($\alpha+\beta$)/2}、に含まれれば、左のサブツリーに挿入され(512,513～515)、そうでなけ

search(condition of last half of 404).

It decides (405) whether or not rectangular which node has concerninganything except node which makes out of range with decision 404 thedescription above, is included in output range [X1,Y1] x [X2,Y2]. If decision of namely, 405 is unattained , because with out of range , the node where on one hand, same decision is formed is inside range,you keep (406) as object of output.

Consequently, push (Input) it does present search object node and segment [;al and;be]with stack system and (407), calculating [;al and;be] segment , traverse it makes node on left and (408 and 409), it returns to decision 403.

At time of this , decision of 403 is formed, is not a node and if it is decided, that it is a empty node , stack decides theempty whether or not with 410. When being decided that stack is not sky, (410), [poppu] it does the stack and (411), traverse it makes node right and (412 and 413), itreturns to decision 403, repeats decision and operation after 404. Result of decision 403 being affirmative , there is not a node and (With empty node),furthermore result of decision 410 and stack are sky with the affirmative , if it is, ends because it means that searching concerning the all desire node , treatment is stopped.

Consequently, referring to flowchart which is shown in Figure 5 , youexpress concerning formation and additional treatment of the tree *node . n, ;al, the;be and x, y etc are similar to supposition incase of aforementioned search process . In addition in order to actualize node formation operation moreeasily, here start treatment from node of dummy which has the root *node on left.

It designates node which it should insert newly as new and(501), following to construction of tree , it traces each node . namely, Pn is done dummy node (Figure 1 10) with and (502), left node of the Pn set is designated as n, (503). flag value which shows "Left" in in is done set and(504), (0, N) set is done in segment (;al and;be), (505).

If with following decision 506, n check does empty node whether or not and it is not a empty node , it decides 507. In this decision 507, if lower left xco-ordinate x1 (n) of rectangular of present search object node , is large in comparison with lower left xco-ordinate x1 (new) of rectangular of the node which it should insert, because location which it should insertis location of reality node , present node to attach each pointer of node which it should insert and to exchange both node ,by changing. (508 and 509). It makes node which should insert present node (510).

Furthermore, if top right xco-ordinate x2 (new) of rectangular of node which it shouldinsert [;al, (;al +:be) / 2], is included, it is inserted in sub tree left and (512,513 - 515),is not so, it is

れば右のサブツリーに挿入される(512,516～518)。

inserted in sub tree right (512,516 - 518).

最後に、挿入すべきノードは、必ずツリーの葉として生成される(519)ので、その左右の子ノードを空ノードとして(520)処理を終了する。

Because lastly, node which it should insert is formed (519), by allmeans as leaf of tree it ends (520) treatment with child node on left and right as empty node .

ツリー・ノード操作処理の他の例として、ノードの削除処理について、第5図のフローチャートを参照して説明する。n, α, β, x1, y1 等は、前記仮定と同様である。また、ダミーのノードより処理を開始することも、前述と同様である(602～605).

As other example of tree *node operating process , concerning deletion processing of node ,referring to flowchart of Figure 5 , you explain. n, ;al, the;be and x1, y1 etc are similar to theaforementioned supposition , In addition, also, it is similar to earlier description from node of dummy to start treatment, (602 - 605).

削除すべきノードを前述の検索処理によって検索、確定し、これを del とする(601)。del とされたノードは木ツリーは、前に詳述した定義と手法に従って構成されているため、ノード生成時と同様の手法で、その構造に従ってツリーをたどることにより(606～614)、del の位置(del の親ノード Pn、左右区別 dn)を探索することができる。

With aforementioned search process it searches, decides node which itshould delete, designates this as del (601). As for node which makes del as for this tree , following todefinition and technique which are detailed before, because configuration it is done. with technique which is similar to time of node formation, following to construction , by tracing tree (606 - 614), you cansearch location (Parent node Pn, left and right distinction dn of del) of del.

delが発見された(606)ならば、delをツリーより削除する。すなわち、ツリーの構造を保つために、下記のようにして、del の子(左,右)ノードによりそのノードを置き換える。del の右ノードが無い(空ノード)か(616)、または、del の左ノードの矩形左下 x 座標が、右ノードの矩形左下 x 座標よりも小さい場合(618)は、del の左ノードで置き換える。この場合の置き換えは、del の親ノードに del の左ノードを接続(626)し、del の右ノードを左ノードの右ノードとし(625,630)、del に del の左ノードのポインタを保持させること(629)によって行なう。

If it is a (606) where del is discovered, del is deleted from tree . construction of namely, tree was maintained, *. description belowrequiring, it replaces node with child (Left and right) node of del. There is not right node of del, when (Empty node) (616), or, rectangular lower left xco-ordinate of left node of del, it is small in comparison with the rectangular lower left xco-ordinate of right node , it replaces (618), with left node of del. Replacement in this case connection (626) does left node of the del in parent node of del, designates right node of del as right node of left node and (625 and 630), does with thething (629) which keeps pointer of left node of del in the del.

また、del の右ノードで置き換える場合(617,618)も同様に行なう(619～624)。

In addition, when it replaces with right node of del, itdoes also (617 and 618) in same way, (619 - 624).

このようにして、削除すべきノードを葉に向って一段一段置き換えてゆき(615)、葉ノードを削除したところで(631)、この処理は終了する。

this requiring, one step one step it replaces node which it shoulddelete facing toward leaf and (615), being a place where leaf node is deleted. (631), this treatment ends.

つぎに、第 1 図に示す図形情報をもとに、前述の各編集操作、すなわち、図形情報の新規生成、追加及び削除の各操作を通しての、ツリーの状態の経緯を具体的に述べる。

Next, warp and weft of state of tree is expressed on basis of graphic shape data which is shown in Figure 1 , concretely through eachoperation of aforementioned each compilation operation, novel formation, addition and deletion of namely, graphic shape data .

まず、図形情報を新たに生成する場合についての例を、第7図および第5図を参照して述べる。新規に図形情報を生成する場合は、まず、ダミーのツリー・ノード 10 を生成しておく(701)。その後、第 1 図②の図形情報が操作者により入力(例えば、マウス 14 等で多角形の各頂点をグラフィック・ディスプレイ上でクリックすることにより)されたと仮定すると、その図形情報より外接矩

First. when graphic shape data is formed anew, being attached, example.referring to Figure 7 and Figure 5 , you express. When graphic shape data is formed in novel , first, tree *node 10 of dummy is formed, (701). After that, that graphic shape data of Figure 1 2 it was inputted (Each apex of polygonal shape on graphic *display in with such as for example mouse 14 click doing to depend) by the operator when supposition it does, circumscribed rectangular

形座標{x1(2),y1(2)}，{x2(2)，y2(2)}を求めて、新たなノード 3 に格納する。

co-ordinate {x1 (2), y1 (2) }, seeking the {x2 (2), y2 (2) } from graphic shape data , it houses in new node 3.

この後、第 5 図に示すフローチャートに従い、格納したノードを new(-②のノード)とし(501)、Pn を第 1 図のダミーノード 10 として(502)、ツリーの生成を行なう(702)。

new (- node of 2) with it does node which is housed and after the this , in accordance with flowchart which is shown in Figure 5 , (501),with Pn as dummy node 10 of Figure 1 (502), it forms tree , (702).

この場合は、Pn を(ダミーノード)の左、右のノードは存在せず、空ノードとなっているため、n には空ノードがセットされ(503)、504-505 の処理後、直ちに ノード挿入処理、Pn(-ダミーノード)の in(-左)ノードに new(-②のノード)をセットし(519)(703)、new の左、右ノードを"空ノード"にセットする(520)(704)を行なう。以上で、図形情報②の新規生成を行なう手順は終了する。

In case of this , Pn node on left and right of(dummy node) does not exist, because it becomes empty node , empty node set is done to n and (503), after treating 504 - 505,set does new (- node of 2) at once in in (- Left) node of node insertion process . Pn (-dummy node) and (519) (703), left of new, right node does (520) (704)which " set is made empty node ". At above, protocol which forms graphic shape data 2 novel ends.

次に、前記状態より第 1 図に示した③,⑤, ①の図形情報がこの順に入力され、図形情報を追加する場合について、各々第 8,9,10 図を参照して説明する。

Next, graphic shape data of 3, 5 and 1 it shows in Figure 1 , is inputted by this order from aforementioned state when graphic shape data is added,being attached, referring to each 8 th , 9, 10 figures, explains.

③

3

の図形情報が入力されて、外接矩形座標を生成した後、新たなノードに格納する。この後、第 5 図に示すフローチャートと第 8 図に示す経緯図に従い、③を格納したノードを new とし(501)、Pn を第 1 図のダミーノード 10 として(502)、ツリーの生成を行なう(802)。

graphic shape data being inputted, after forming circumscribed rectangular co-ordinate , ithouses in new node . It designates node which houses 3 after this , in accordancewith warp and weft figure which shows in flowchart and Figure 8 which areshown in Figure 5 , as new and (501), with Pn as dummy node 10 of Figure 1 (502), it forms tree , (802).

この場合は、Pn の左ノードにはルート・ノード(②のノード)がセットされているため、n には②のノードがセットされる(503)。504～505 の処理後、n すなわちノード②は空ノードではない(506)ので、x1(new){-x1(3)}と x1(n){－x1(2)}とを比較する(507)。

In case of this , because root *node (node of 2) set is done to theleft node of Pn, node of 2 set is done to the n, (503). After treating 504 - 505, as for n namely node 2 it is nota empty node , because (506), x1 (new) {-x1 (3) } with x1 (n) {-x1 (2) } iscompared (507).

x1(new)＜x1(n)ではないので、Pn に n(-②のノード)をセットし(511)、その後 x2(new)と($\alpha+\beta$)/2(-N/2)を比較する(512)。x2(new)≦($\alpha+\beta$)/2 であるので、n の左のノード(空ノード)を n とし(513)、[α,β]に[α,($\alpha+\beta$)/2](－[0,N/2])とセットして(514)、in に左を示すフラグ値をセット(515)する(803)。

Because x1 (new) <x1 it is not a (n), n (- node of 2) set isdoue in Pn and (511), after that x2 (new) with (;al +;be) / 2 (-N/2) arecompared (512). Because x2 (new) <= (;al +;be) / 2 is, node (Empty node) on left of the n is designated as n, (513), [;al, (;al +;be) / 2] (- [0, N/2]) with set making the[;al and;be], (514), flag value which shows left in in set (515)(803).

続いて、n には空ノードがセットされていることが判る(506)ので、ノード挿入処理(519,520)を行なう(804, 805)。ノード挿入後は、描画順を保持するため、②を格納したノードに、③のノード・ポインタ(②の次に③が描画されていることを示す。)をセットし、一方、③を格納したノードには、②のノード・ポインタ(③の前に②が描画されていることを示す。)をセットする(805)。

Consequently, it understands that empty node set is done to n. because (506), node insertion process (519 and 520) is done. (804 and 805). After node insertion, in order to keep drawing order, in node which houses 2, node *pointer (Next 3 of 2 shows fact that drawing it is done.) of 3 is done set , on one hand, the node *poiuter (3 ago 2 shows fact that drawing it is done.) of 2 set is done in node which houses 3. (805).

これにより、図形情報を描画順に表示すること

Because of this, it becomes possible, to indicate graphic shape

が可能となり、図形情報が重なるような位置に入力され、描画されても、入力した通りに表示することが可能となる。例えば、赤色に塗りつぶした長方形の上に、白色で文字 A,B,C 等と描画し、これらの文字を強調する等のことが可能になる。続いて、⑤の図形情報を入力した場合のノードの挿入は、③を入力した場合と同様に、第5図に示すフローチャートと第9図に示す経緯図に従って行なわれる。

すなわち、⑤のノードを new として(501)、ダミーのノードよりツリー生成を開始する(502～505)(901,902)。n は②のノードであり、判定 506 は不成立であるので、x1(new){-x1(5)}と x1(n){-x1(2)}とを比較する(507)。x1(new)<x1(n)ではないので、Pn に n をセットし(511)、x2(new){-x2(5)}と $(\alpha+\beta)/2(-N/2)$を比較する(512)。

$x2(new)>(\alpha+\beta)/2$、すなわち前記判定 512 は否定であるので、n の右ノード(空ノード)に n をセットし(516)、$[\alpha,\beta]$に$[(\alpha+\beta)/2+1,\beta]$(-[N/2+1, N]とセットして(517)、in に右を示すフラグ値をセット(518)する(903)。

続いて、n に空ノードがセットされていることが判る(506)ので、ノード挿入処理(519,520)を行なう(904,905)。ノード挿入後は、描画順保持のためのポインタを生成する。即ち、③のノードに、⑤のノード・ポインタを格納する。その結果、第9図に 906 で示したように、③には、③の前の図形情報を示す②のノード・ポインタと、③の次の図形情報を示す⑤のノード・ポインタの 2 つが保持される。また、⑤のノードには、③のノード・ポインタをセットする(906)。

①

の図形情報に対するノードの挿入は、前述した図形情報③、⑤の場合と同様に、第 5 図に示すフローチャートと第 10 図に示す経緯図に従って行なわれる。

①のノードを new として(501)、ダミーのノードよりツリー生成を開始する(502～505)(1001,1002)。ここで、n は②のノードであるから空ノードではなく(506)、x1(new){-x1(1)}<x1(n){-x1(2)}ので(507)、new の左、右ノードに各々n の左ノード(③のノード)、右ノード(⑤のノード)をセットする(508)(1003)。

この段階では、new の左ノードは③のノード、右ノードには、⑤のノードがセットされている。次に、Pn(-ダミーノード)の in(－左)ノードに new(－①のノード)をセットする(509)(1004)。そして、n と new とを交換する(510)(1005)。この場合、n は①

data in drawing order, is inputted by kind of location where graphic shape data is piled up,drawing is done, as inputted, it becomes possible to indicate. On rectangle which is painted in for example red color , character A, B, C etc and the drawing it does with white , or other thing which emphasizes these character becomes possible. Consequently, insertion of node when graphic shape data of 5 is inputtedis done in same way as case where 3 is inputted, following to warp and weft figure which shows in flowchart and Figure 9 which areshown in Figure 5 .

With node of namely, 5 as new (501), tree formation isstarted from node of dummy (502 - 505) (901 and 902). Because as for n with node of 2, as for decision 506 itis a unattained , x1 (new) {-x1 (5) } with x1 (n) {-x1 (2) } is compared (507). Because x1 (new) <x1 it is not a (n), n set isdone in Pn and (511), x2 (new) {-x2 (5) } with (;al +;be) / 2 (-N/2) are compared(512).

- [N/2+1, N] With set doing, (517), flag value which shows rightin in set (518) (903). Because x2 (new) > (;al +;be) / 2, namely aforementioned decision 512 is negative , n set is done in right node (Empty node) of the n, (516), in [;al and:be] [(;al +;be) / 2 + 1,:be]

Consequently, empty node set being done you understand in n, because (506), node insertion process (519 and 520) is done, (904 and 905). After node insertion. pointer for drawing sequential retention isformed. Namely, in node of 3, node *pointer of 5 is housed. As a result, as in Figure 9 shown with 906, node *pointer of 2 it shows graphic shape feeling strong of 3 ago and two of node *pointer of 5 thefollowing graphic shape data of 3 is shown are kept in 3. In addition, node *pointer of 3 set is done in node of 5, (906).

1

Insertion of node for graphic shape data is done in same way as thecase of graphic shape data 3, 5 which is mentioned earlier, following to warp and weft figure which shows in flowchart and Figure 10 which are shown in the Figure 5 .

With node of 1 as new (501), tree formation is started from node of dummy (502 - 505) (1001 and 1002). Because here, n is node of 2, not to be empty node ,because (506), x1 (new) {-x1 (1) } <x1 (n) {-x1 (2) } (507), left of new,left node of each n (node of 3), right node (node of 5) set is donein right node , (508) (1003).

With this step , as for left node of new node of 5 the set is done, to node , right node of 3. Next, new (- node of 1) set is done in in (- Left) node of the Pn (-dummy node), (509) (1004). And, n and new are exchanged (510) (1005). In case of this , n becomes node of 1, new becomes node of 2. In

のノードとなり、new は②のノードとなる。つまり、ツリーの途中の位置にノードが挿入された場合は、挿入すべきノードを交換して、③や⑤のノード挿入時と同様にツリーをたどる。

即ち、Pn に n(－①のノード)をセットし、x2(new){－x2(2)}≦(α＋β)/2(-N/2)であるので(512)、n に n の左ノード(③のノード)、[α,β]に[α,(α＋β)/2](－[0,N/2])、in に左を示すフラグ値をセットする(513～515)(1006). 更に、同様にして、n(－③のノード)の判定を行なう(506). x1(new){-x1(2)}<x1(n){－x1(3)}であるから(507)、new(-②のノード)の左、右ノードに n の左、右ノード(ともに空ノード)を各々セットし(508)(1007)、Pn(－①のノード)の in(－左)ノードを new とセットして(509)、n と new の交換をする(510)(1008)。ここで new－③のノード、n－②のノードとなっている。

続いて、Pn に n(－②のノード)をセットする(511)(1009). x2(new){-x2(3)}≦(α＋β)/2(-N/4)であるため(512)、n に n のとnoleft ノード(空ノード)を、[α,β]に[α,(α＋β)/2](-[0,N/4])を in に左を示すフラグ値をセットする(513～515)(1010).

n は空ノードであるので(506)、Pn(－②のノード)の in(-左)ノードに、new(-③のノード)をセットし(519)(1011), new の左右ノードを空ノードにセットする(520)(1012)。

上述のツリー生成処理後は、描画順ポインタを、③,⑤の入力時と同様の手順で、①のノードと、⑤のノードにセットする(1013)。

以上に詳細に述べたように、本発明では、ツリーの生成時には、ノードごとに2分された枝を、1つのパスしか通らない、即ち、1 つのノード挿入のためには、ツリーのルートより葉までの一本道しかたどらないため、たどるノードの数は、全ノード数(すなわち、全図形情報数)を K とした場合、ほぼ log2K でよいことになる。

従って、全図形情報が 100000 程度の多数である場合でも、17 回程度(≒log2 100000)のノード探索で操作を完了することが可能となっている。

具体例の最後として、図形情報を削除する場合について、以下に説明する。

図形情報を削除する場合は、操作者が入力した座標(例えば、マウス等で削除対象となる図形の近傍を、クリックした位置)をもとに、第 4 図で示されるツリー探索処理を用いて、対象図形情報に対応したツリー・ノードを確定し、第 6 図で示されるツリー・ノード削除処理にて、ノードを削除し、更に図形情報も削除する。

other words, when node * entrance it is done in location immiddle of tree , exchanging node which it should insert,it traces tree in same way as time of node insertion of3 and 5.

Namely, because set to do n (- node of 1), x2 (new) {-x2 (2) } <= (;al +;be) /they are 2 (-N/2) in Pn, (512). in n left node of the n (node of 3), [;al, (;al +;be) / 2] (- [0, N/2]). flag value which shows left in in the set is done in [;al and;be], (513 - 515) (1006). Furthermore, it decides n (- node of 3) to similar, (506). Because x1 (new) {-x1 (2) } <x1 it is a (n) {-x1 (3) }, (507), left of the new (- node of 2), left of n, each set it does right node (Together empty node)in right node and (508) (1007), new and set doing in (- Left) node of Pn (- node of 1), (509), it exchanges n and new, (510)(1008). It has become node of node , n- 2 of new- 3 here.

Consequently, n (- node of 2) set is done in Pn, (511) (1009). Because x2 (new) {-x2 (3) } <= (;al +;be) / they are 2 (-N/4), (512), in n theleft node (Empty node) of n, [;al, (;al +;be) / 2] (- [0, N/4]) flag value which shows theleft in in set is done in [;al and;be], (513 - 515) (1010).

Because n is empty node , (506), in in (- Left) node of the Pn (- node of 2), new (- node of 3) is done set and (519) (1011), left and right node of new set is designated as empty node , (520) (1012).

After above-mentioned tree generation , drawing sequential pointer , with the protocol which is similar to when inputting 3 and 5, set isdesignated as node of 1 and node of 5, (1013).

As expressed in detail above, with this invention , when forming tree ,2 is divided only pass of one it passes by branchwhich, to every node . Namely, for node insertion of one , because only one Motomichito leaf it traces from root of tree , quantity of node which is traced, when all node several (Quantities of namely, all graphic shape data) are designated as K, means almost to be possible to be a log2K.

Therefore, 17 times it has become possible even with when all graphic shape data is large number of 100000 to complete operation with node search of (S log2 100000).

embodiment as last, when graphic shape data is deleted, being attached, youexplain below.

When graphic shape data is deleted, co-ordinate (click does vicinity of graphic shape which becomes deletion object with such as for example mouse , location which) which operator inputs on thebasis of, it decides tree *node which corresponds to object graphic shape data makinguse of tree search treatment which is shown with Figure 4 , with tree *node deletion processing which is shown with Figure 6 , it deletes node ,furthermore deletes also graphic shape data .

第 1 図に示されたツリー及び図形情報が記憶されており、②の図形情報のひし形の右下辺上の座標が、削除対象として与えられた場合に、②の図形情報を削除するまでの経緯についての例を、以下に述べる。

When tree and graphic shape data which are shown in Figure 1 areremembered, co-ordinate on right bottom side of diamond shape of graphic shape data of2, are given, as deletion object until graphic shape data of 2 is deleted,example concerning warp and weft , is expressed below.

まず入力された座標を(x.y)とし、N/4＜x≦N/2であり、かつ x1(4)＜x＜x2(4)であると仮定する。更に、ε を N に比較して十分小さい数として(例えば、N=1024 とすれば ε=8 程度)、X1=x－ε 、Y1=y－ε ,X2=x+ε , Y2=y+ε 、とする。

First when co-ordinate which is inputted is done (x, y) with, with the N/4

これらの条件をもとに、まず第 4 図に示すフローチャートに従い、第 1 図で示されるツリーをたどる例について示す。更に、どのようにツリーがたどられるかの軌跡を、第 11 図に示す。

These condition on basis of, it shows concerning example whichtraces tree which is shown with Figure 1 in accordance with the flowchart which is shown first in Figure 4 . Furthermore, which way trajectory whether tree is traced of, isshown in Figure 11 .

第 11 図において、2 重丸のノード①,②,④は、図形情報が削除対象か否かを検索したノード、1 重丸のノード③,⑤は、ツリートラバースの過程で探索したノードである。また、三角枠のノード⑥,⑦は、探索しないノードである。また、1108 は、ノードをたどる順を表す矢線である。まず、第 4 図のフローチャートにおいて、n をルート・ノード(①のノード)とし(401)、[α,β]を[0,N]とする(402)。n はルート・ノードであって空ノードではなく(403)、しかも X2＞x1(n){－x1(1)}かつ X1＜β (－N)であるので(404)、[X1,Y1]×[X2,Y2]と[x1(n),y1(n)]x[x2(n), y2(n)](n-1)との領域が重なるか否かの判定を行なう(405)。これらの領域が重ならないので、スタックに n(-1)、[α,β](-[0,N])をプッシュして保持し(407)、n に n(－①のノードの)の左ノード(②のノード)をセットする(408)。そして、次のノードに対応する区間をセットするために、[α,β]に[α,(α+β)/2](-[0,N/2])をセットして(409)、判定 403 へ戻る。

It decides whether or not X1, Y1*X [X2, Y2] with [x1 (n), y1 (n)] x [x2 (n), y2 (n)] (n- 1) with region is piled up of, (405). Because these region are not piled up, n (- 1), push doing[;al and;be] (- [0, N]) in stack , you keep and (407), set you do left node (node of 2) of n (- node of 1) in n, (408). And, in order set to do segment which corresponds to thefollowing node , set doing [;al, (;al +;be) / 2] (- [0, N/2]) in [;al and;be], (409), it returnsto decision 403. With Figure 11 snell, as for node 1, 2, 4 of double circle, as for the node , 1 Shigemaru where graphic shape data searches deletion object whether or not node 3, 5, itis a node which is searched with process of tree traverse . In addition, node 6, 7 of triangle framework is node which isnot searched. In addition, 1108 is arrow which displays order which traces node . First, n root *node (node of 1) with is done and in flowchart of the Figure 4 , (401), [;al and;be] is done [0, N] with, (402). As for n not to be a empty node with root *node , (403),furthermore because X2>x1 (n) {-x1 (1) } and it is a X1<;be (-N), (404),

n(－②のノードは空ノードではなく(403)、X2＞x1(n){-x1(2)}かつ X1＜β (-N/2)であり(404)、[X1, Y1]x[X2, Y2]と[x1(n), y1(n)]x[x2(n), y2(n)](n-2)との領域が重なる(405)ので、n で示されるノードは、検索されたノードである(406)。

2 >} and (404), [X1,Y1] x [X2,Y2] with [x1 (n), y1 (n)] x [x2 (n), y2 (n)] (n- 2) with region ispiled up with X1<;be (-N/2), because (405), node which is shownwith n is node which is searched, (406). -x1 As for n (- node of 2) not to be a empty node , (403), X2>x1 (n)

更に同様にして n,[α,β]をスタックにプッシュして保持し(407)、n に n の左ノード(③のノード)とにセットし(408)、[α,β]に[α, (α+β)/2](－[0,N/4])とした後、再び判定 403 へ戻る。

After [:al, (;al +;be) / 2] (- [0, N/4]) with making the;al and the;be), again it returns todecision 403. Furthermore push designating n, [;al and;be] as stack tosimilar, you keep and (407), left node of n (node of 3) with the set you make n, (408),

n(-③のノードは空ノードではなく(403)、X1>β (-N/4)であり(404)、またスタックは空でない(410)ので、スタックをポップして n、区間を取り出し n,[α,β]にセットする(411)。この時の n は②の

As for n (- node of 3) not to be a empty node , (403), (404), in addition the stack is not sky with X1>;be (-N/4), because (410), stack [poppu],it removes node , segment and set makes n, [;al and;be] (411). As for n at time of this as for node , [;al

ノード、[α,β]は[O,N/2]である。

and;be] of 2 itis a [O, N/2].

n に n の右ノード(④のノード)をセットし(412)、[α,β]に[(α+β)/2+1,β](ー[N/4+1,N/2])をセットする(413)。

Right node (node of 4) of n set is done in n and (412),[(;al +;be) / 2 + 1,;be] (- [N/4+1, N/2]) set is done in [;al and;be], (413).

更に、判定 403 へ戻ってツリーをたどる。このときも n(ー④のノード)は空ノードではなく(403)、X2＞x1(n)(ーx1(4))かつ X1＜β(-N/2)であり(404)、[X(sub)1、Y1]×[X2, Y2]と[x1(n),y1(n)]x[x2(n), y2(n)](n-4)とは領域が重ならない(405)。

-x1 (4) and (404), [X (sub) 1, Y1] X [X2,Y2] with [x1 (n), y1 (n)] x [x2 (n), y2 (n)] (n- 4) with the region is not piled up with X1<;be (-N/2), (405). Furthermore, returning to decision 403, it traces tree . At time of this as for n (- node of 4) not to be a empty node ,(403), X2>x1 (n)

それ故に、スタックに n,[α,β]をプッシュして保持し(407)、n に n の左ノード(空ノード)をセットし(408)、[α,β]に[α,(α+β)/2](ー[N/4+1,(N/4+1+N/2)/2])をセットして(409)、判定 403 へ戻る。

Therefore, push doing n, [;al and;be] in stack , you keep and (407),set you do left node (Empty node) of n in n, (408), the set doing [;al, (;al +;be) / 2] (- [N/4+1, (N/4+1+N/2) / 2]) in [;al and;be], (409), it returns to decision 403.

次の判定では、n の空ノードであり(403)、スタックは空でないので(410)、スタックをポップしてノード、区間を取り出し、n,[α,β]にセットする(411)。この時の n は④のノードであり、[α,β]は、[N/4+1,(N/4+1+N/2)/2]である。

Because with following decision, (403), as for stack it is not asky with empty node of n, (410), stack [poppu], it removes node , segment , set makes n, [;al and;be] (411). As for n at time of this with node of 4, as for[;al and;be], it is a [N/4+1, (N/4+1+N/2) / 2].

n に n の右ノード(空ノード)をセットし(412)、[α,β]に[(α+β)/2,β](ー[(N/4+1+(N/4+1+N/2)/2)2+1,(N/4+1+N/2)/2])をセットして(413)、判定 403 へ戻る。

set doing [(;al +;be) / 2,;be] (- [(N/4+1+ (N/4+1+N/2) / 2) 2 + 1, (N/4+1+N/2) / 2]) in the;al, and the:be] (413), it returns todecision 403. Right node (Empty node) of n set is done in n, (412),

この場合も、n は空ノードであり(403)、スタックは空でないので(410)、更にスタックをポップしてノード、区間を取り出し、n,[α,β]にセットする(411)。この時の n は①のノード、[a,β]は[0、N]である。

In case of this , because as for n (403), as for stack itis not a sky with empty node , (410), furthermore stack [poppu], itremoves node , segment , set makes n, [;al and;be] (411). As for n at time of this as for node , [a, ;be] of 1 itis a [0, N].

n を n の右ノード(⑤ノード)とし(412)、[α,β]を[(α+β)/2+1,β](ー[N/2,N])とする(413)。その後、判定 403 へ戻る。

n is done right node of n (5 node) with and (412), [;al and;be]is done [(;al +;be) / 2 + 1,;be] (- [N/2, N]) with, (413). After that, it returns to decision 403.

n(ー⑤ノード)は空ノードでなく(403)、X2＜X1(n)(ーX1(5))であり(404)、スタックは空(410)であるので、ツリー探索を終了する。

Because -X1 (404), stack is empty (410) with (5), it ends tree search. As for n (- 5 node) not to be a empty node (403), X2

以上の処理によって、削除対象の図形情報は、②のノードに対応したものであることが分る。ここで、実際に検索されたノードは、下記の通りとなる。

In treatment above, as for graphic shape data of deletion object , itunderstands that it is something which corresponds to node of 2. Here, node which is searched actually becomes. as descriptionbelow.

①,②,④:

1, 2 and 4:

対象の図形情報か否かのチェックを行なったノード。

node . which did check of graphic shape data whether or not of object

③,⑤:

3 and 5:

ツリートラバースの過程として検索したノード。

node . which it searches as process of tree traverse

これより、対象の図形情報か否かの検索の為には、ノードごとに2分された枝の、1つのパスしか通らないことが分る。即ち、ツリー生成時と同様に、その検索対象のノード数は、全ノード数

From this, for searching graphic shape data whether or not of object , 2 is divided, only pass of one of branch which it understands in every node that it does not pass. Namely, in same way as time of tree formation, as for thequantity of node

(全図形情報数)をKとした場合ほぼlog2K程度であることが分る。

of search object , when all node several (Quantities of all graphic shape data) aredesignated as K, it understands that it is a log2K extentalmost.

従って、従来技術では、全図形情報が、たとえば100000個あった場合は、同数の100000回図形情報を検索する必要があったが、本方式では、わずか17回程度(≒log2 100000)で済む。これに、ツリートラバースのオーバーヘッド、即ち、ツリートラバースの過程として検索したノードの個数を加えても、その検索回数には、大きな開きがあることが判る。

Therefore, with Prior Art , when all graphic shape data , for example 100000 it is. it wasnecessary to search 100000 time graphic shape data of same number , but with this system . only 17 times is sufficient (S log2100000). In this, as overhead , of tree traverse namely process of tree traverse including number of node which it searches, it understands insearching number of times that it is opening which is large.

次に、削除対象となった図形情報の②のノードを、ツリーから削除する具体例について、第6図に示されたフローチャートと第12図の経緯図を参照して説明する。

Next, referring to warp and weft figure of flowchart and Figure 12 whichare shown in Figure 6 , concerning embodiment which deletes node of 2 of graphic shape data which has become deletion object , from tree ,you explain.

まず、delを②のノードにセットする(601)(1201)。Pnを第1図のダミーノード10とし(602)、nにPnの左ノード(ルート・ノードである①のノード)をセットし(603)、dnに「左」を示すフラグ値をセットし(604)。区間[α,β]に[0, N]をセットする(605)(1202)。

First, del set is designated as node of 2, (601) (1201). Pn is designated as dummy node 10 of Figure 1 and (602), left node (** * * node of 1 it is a node) of Pn set is done in n and (603), flag value which shows "Left" in dn is done set and (604), [0, N] set is done in segment [;al and;be], (605) (1202).

つぎに、n(ー①のノード)は del(ー②のノード)ではないので(606)、Pnにn(-①のノード)をセットする(607)。x2(del)(-x2(2))≦($\alpha+\beta$)/2(ーN/2)であるので判定608は肯定であり、nにn(ー①のノード)の左ノード(-②のノード)をセットする(609)。

Because -x2 (2) <= (;al +;be) / they are 2 (-N/2), decision 608 with affirmative ,set does left node (- node of 2) of n (- node of 1) in n, (609). Because next, n (- node of 1) is not del (- node of 2), (606), n (- node of 1) the set is done in Pn, (607). x2 (del)

[α,β]に[α,($\alpha+\beta$)/2](-[0,N/2])をセットし(610)、dnに「左」を示すフラグ値をセットする(611)(1203)。このとき、n(-②のノード)は delであり(606)、delの左、右ノードはともに空ノードでない(616～617)ので、x1(delの左ノード){-x1(3)}とx1(del の右ノード)(-x1(4))とを比較する判定618を実施する。

-x1 decision 618 which compares (4) is executed. [;al, (;al +;be) / 2] (- [0, N/2]) set is done in [;al and;be] and (610), flag value whichshows "Left" in dn is done set , (611) (1203). At time of this , as for n (- node of 2) (606), left of the del, as for right node it is not a empty node together with del. because (616 - 617), x1 (Left node of del) {-x1 (3) } with x1 (Right node of del)

ここでは、x1(delの左ノード)<x1(delの右ノード)であり、判定618が成立するので、delの右ノード(④のノード)をnにセットし(625)(1204)、Pn(-①のノードの dn(左)ノードに delのノード(③のノード)をセットする(625)(1205)。del のノード(③のノード)を Pnにセットし(627)(1206)、dnに「左」を示すフラグ値をセットする(628)。delの左、右ノードに Pn(-③のノードの左、右ノード(ともに空ノード)をセット(629)(1207,1208)、Pn(-③のノードの右ノードに n(-④のノード)をセットする(630)。上述の処理操作を行なうことによりツリー・ノードの再構成を行なうことができる(1209)。この後、判定615へ戻るが、このときは del(-②のノード)の左、右ノードはともに空ノードとされているので(615)、Pn(-③のノードの dn(左)に空ノードをセットし(631)、ノードの削除処理を完了する(121

Because here, x1 (Left node of del) <x1 with (Right node of del), decision 618 is formed.right node (node of 4) of del set is designated as n and (625) (1204), left node (node of 3) of del set is done in the dn (Left) node of Pn (- node of 1). (625) (1205). Left node (node of 3) of del set is designated as Pn and (627) (1206), flag value which shows "Left" in dn is done the set , (628). Left of del, in right node left of Pn (- node of 3),right node (Together empty node) set (629) (1207 and 1208), n (- node of 4) set is done inright node of Pn (- node of 3). (630). It is possible to do reconstruction of tree *node , by doing theabove-mentioned processing operation (1209). After this , it returns to decision 615, but because at timeof this left of del (- node of 2), as for right node it ismade empty node together, (615), empty node set is donein dn (Left) of Pn (- node of 3) and (631), deletion processing of node iscompleted (1210).

0)。

そしてさらに、描画順序保持のためのポインタを削除する。即ち、②の図形情報の直前に入力された図形情報(この場合は存在しない。)の次には、②の図形情報の直後に入力された図形情報(③の図形情報)が並ぶようにポインタをセットし、②の図形情報の直後に入力された図形情報(③の図形情報)の前には、②の直前に入力された図形情報(この場合は存在しない。)が並ぶようにポインタをセットする(1211)。

And furthermore, pointer for drawing order retention is deleted. Namely, next of graphic shape data (In case of this it does not exist.) which is inputted immediately before the graphic shape data of 2, in order for graphic shape data (graphic shape data of 3) which is inputted immediatelyafter graphic shape data of 2 to line up pointer is done set , in orderfor graphic shape data (In case of this it does not exist.) which is inputted immediately before 2 to line up the pointer set is done before graphic shape data (graphic shape data of 3) which is inputtedimmediately after graphic shape data of 2 (1211).

以上で、図形情報の削除を行なう手順は終了する。

At above, protocol which deletes graphic shape data ends.

以上述べたように、ツリーよりのノード削除はノードごとに 2 分された枝を、1 つのパスしか通らない。即ち、ツリー生成と同様の理由より、そのたどるノード数は全ノード数(全図形情報数)を K とした場合、ほぼ log2K であることがわかる。

As above expressed, node deletion from tree 2 is divided only pass of one passes by skill which, to every node . Namely, as for quantity of node which that is traced from thereason which is similar to tree formation, when all node several(Quantities of all graphic shape data) are designated as K, it understands that it is a log2K almost.

本システムの会話形図形編集プログラムは、操作者によって起動され、図形情報の編集指示を受けた場合、全図形情報の中から編集範囲の図形情報を前記ツリー探索処理で高速に検索し、グラフィック・ディスプレイに表示する。

conversation shape graphic shape compilation program of this system is started with operator , when compilation display of graphic shape data is received, frommidst of all graphic shape data in aforementioned tree search treatmentsearches graphic shape data of compilation range in high speed , indicates in the graphic *display .

なお、表示の際、図形情報が重なっており、描画順に表示図形が依存しているような場合は、検索されたノード・ポインタを適当な配列に保持し、描画順にソートした後、表示する。更に、削除による表示画面の修復にも使用できる。

Furthermore, when indicating, graphic shape data is piled, indicator graphic shape depends on drawing order is, kind of when, you keep node *pointer which issearched in suitable arrangement, after sort making drawing order, youindicate. Furthermore, you can use for also rejuvenation of display screen withdeletion.

また、図形を描画するなどして、新たな図形情報の生成を行う場合は、新たな図形情報に基づいて外接する矩形座標を演算し、ノード生成処理を用いて図形情報の生成を行う。

In addition, when drawing such as does graphic shape doing, forms the new graphic shape data , calculating rectangular co-ordinate which is circumscribed on basis of new graphic shape data , it forms graphic shape data making use of node generation .

既に生成されている図形情報を削除する場合は、適宜の手段で、指定された座標近傍の図形情報を、前記ツリー探索処理を行って、高速に検索する。この場合は、指定された座標位置を含む小さな矩形(たとえば、全空間を[0,1024]x[0,1024]とし、指定座標位置をx,y とした場合、[x-8,y-8]x[x+8,y+8]で定義できる)を用いて、候補図形を検索することができる。

When graphic shape data which is formed already is deleted, graphic shape data of the co-ordinate vicinity which with appropriate means , is appointed, doingaforementioned tree search treatment, it searches in high speed . In case of this , candidate graphic shape can be searched making use of thesmall rectangular ([0 and 1024] x [0 and 1024] with it does for example all space , when designated coordinate position is designated as x, y, it defines with [x-8, y-8] x [x+8, y+8]) which includes coordinate position which is appointed.

また、更に検索精度を上げるためには、複数の候補図形に対して指定座標点との間の距離計

In addition, in order furthermore to increase retrieval precision , itadministers distance gauge calculation between designated co-ordinate pointsvis-a-vis candidate graphic

算を施し、優先順位を付けることもできる。

この実施例によれば、膨大な図形情報の度重なる検索をきわめて高速に行うことができ、操作者に対して快適な編集作業環境を提供できる。

また、LBP やプロッタに対し図形情報を出力する際に応用できることは明らかであり、膨大な図形情報の中の任意の範囲を高速に選択、出力することが可能である。

更に、図形情報を会話形式で入力せず、カードやフロッピーディスク等の媒体を通して、大量の図形情報を一括して入力させることも、ツリー生成時間(入力情報数を K とすると、ほぼ Klog2K に比例した時間)が十分に短かいので、可能である。

また、第 3 図に示すように、2 次記憶装置 21 の容量が不足しており、ツリーを格納できない場合でも、会話編集開始時に主記憶上で、ツリー生成プログラム 22 によってツリーを構成し、編集操作時の応答性を確保することもできる。

本実施例で説明したツリーの構造は、矩形の x 座標に着目したものであるが、全図形情報の座標系が[0,N]x[0,M]であり、M>>N の場合は、明らかに、y 座標に着目してツリーを構成すべきである。このようにすれば、ツリーのレベルを低く抑えることができる。また、座標系[0,N]x[0,M]を任意の実数空間に拡張することも可能である。

(発明の効果)

以上述べたところから明らかなように、座標をキーとして図形情報を検索し、編集出力させる方式に、バイナリ・ツリーを導入したことにより、従来では不可能であった高速応答が可能となった。すなわち、従来の応答時間を kK とした場合、(k+k')log2K に短縮することができる。ただし、上記においては、K は図形情報数、k は検索処理のオーバーヘッド、k'はツリー探索のオーバーヘッドである。

さらに、検索が高速化され、検索に要する時間が短縮されたことにより、大量の図形情報を一元的に管理することが可能となり、その任意範囲の編集、出力が従来技術にて確保されていた応答時間とほぼ変りなく確保できることにより、より高度の操作性を持つことができる。

結果的に、グラフィックス・システムとしての性能

shape of plural , can also attach priority sequence .

According to this Working Example , expansion it is possible, to do searchingwhere graphic shape data occurs repeatedly in quite high speed it can offer comfortable edit task reality boundary vis-a-vis operator .

In addition, when outputting graphic shape data vis-a-vis LBP and the plotter , as for being able to apply being clear. expansion range of the option in graphic shape data it is possible to select and to output to the high speed .

Furthermore, not to input graphic shape data with conversation form , lumping together graphic shape data of large scale through card and floppy disk or other media . becauseinputting, tree production time (When quantity of input information is designated as K, almost it waspproportionate to Klog2K time) is short in fully , it is possible.

In addition, as shown in Figure 3 , with also case, when starting conversation compilation on main memory , configuration it does tree with tree generation program 22, also where capacity of secondary storage 21 is insufficient, the tree cannot house it is possible to guarantee responsiveness at timeof compilation operation.

construction of tree which is explained with this working example is somethingwhich pays attention to xco-ordinate of rectangular , but coordinate system of all graphic shape data being [0, N] x [0, M], when it is a M>>N, clearly, payingattention to yco-ordinate , it is good configuration to do tree . If this it requires, it can hold down level of tree low. In addition. also it is possible to expand coordinate system [0, N] x [0, M] in real number space of option .

(Effect of Invention)

As above been clear from place where you express, graphic shape data wassearched with co-ordinate as key , in system which it compilesoutputs, high speed response which is a impossible until recently by introducing the binary *tree , became possible. When namely, conventional response time is designated as kK, it can shorten to (k+k') log2K. However, as for K quantity of graphic shape data , as for k asfor overhead , k' of search process it is a overhead of tree search indescription above.

Furthermore, searching is done acceleration , it becomes possible tomanage graphic shape data of large scale monistically due to fact that thetime when it requires in searching is shortened, compilation and theoutput of option range are Prior Art and it is possible to havehigher-level operability , by being able to guarantee not to almost bedifferent from response time which is guaranteed.

In resulting , performance as graphics *system , means to

が、飛躍的に向上することになる。

improve rapidly.

Drawings

4.図面の簡単な説明

4.Brief Explanation of the Drawing (s)

第 1 図は本発明の要部であるバイナリ・ツリーの構造と図形情報の概要を示す図、第 2,3 図は、本発明を適用したグラフィックス・システムの概略構成例を示すブロック図、第 4 図は本発明におけるツリーの探索処理を示すフローチャート、第 5 図はツリーの生成処理を示すフローチャート、第 6 図はツリーよりノードを削除する処理を示すフローチャート、第 7 図はバイナリ・ツリーの新規作成時におけるツリー構造の変化状態を示す図、第 8～10 図は図面情報の追加に伴うバイナリ・ツリーのノード追加時におけるツリー構造の変化状態を示す図、第 11 図はバイナリ・ツリーの探索経路を示す図、第 12 図は図面情報の削除に伴うバイナリ・ツリーのノード削除時におけるツリー構造の変化状態を示す図、第 13 図はバイナリ・ツリーを用いた図形情報の探索手順を説明するための座標系を示す図である。

As for Figure 1 as for construction of binary *tree which is a principalpart of this invention and figure and second , 3 figure which show the gist of graphic shape data , As for block diagram , Figure 4 which shows conceptual configuration example of graphics *system whichapplies this invention as for flowchart , Figure 5 which shows search treatment of tree in this invention as for flowchart , Figure 6 which shows generation of the tree from tree as for flowchart , Figure 7 which shows treatment whichdeletes node figure which shows changed state of tree structure at thetime of novel compilation of binary *tree , As for 8 th ~10 figures as for figure and Figure 11 which show the changed state of tree structure at time of node addition of binary *tree whichaccompanies addition of drawing data as for figure and Figure 12 which show search path of binary *tree figure which shows the changed state of tree structure at time of node deletion of binary *tree whichaccompanies deletion of drawing data , Figure 13 is figure which shows coordinate system in order to explain thesearch protocol of graphic shape data which uses binary *tree .

1…

1...

ツリーの全体構成、

entire constitution , of tree

2…

2...

図形情報の全体構成、

entire constitution , of graphic shape data

3…

3...

ツリー・ノード、

tree *node ,

4…

4...

図形情報、

graphic shape data ,

7…

7...

図形情報を囲む外接矩形、

Circumscribed rectangular , which surrounds graphic shape data

8…

8...

ルート・ノードに対応した区間[0,N]、

segment which corresponds to root *node [0, N],

11…

11...

グラフィック・ディスプレイ、

graphic *display ,

12…

12...

キーボード、

keyboard ,

13…

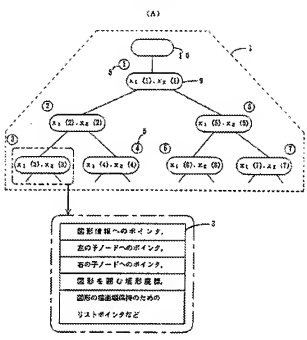
13...

電子計算機、

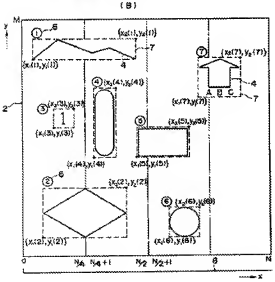
Tatsuko computer ,

14…	14...
マウス、	mouse ,
15…	15...
ライトペン、	light pen ,
15…	15...
デジタイザ、	digitizer ,
17,21…	17, 21...
2 次記憶装置、	secondary storage ,
18…	18...
LBP、	LBP,
19…	19...
プロッタ、	plotter ,
20…	20...
パッチ型入力装置、	patch type input device ,
22…	22...
ツリー生成プログラム	tree generation program
代理人	representative
弁理士 平木道人	patent agent Hiraki Michito

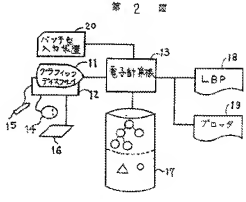
第 1 図 (その1)



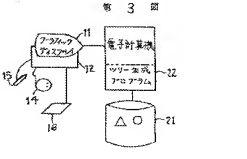
第 1 図 (その2)



第 2 図



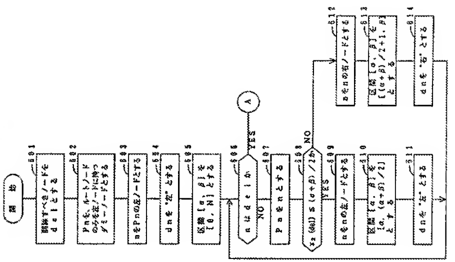
第 3 図



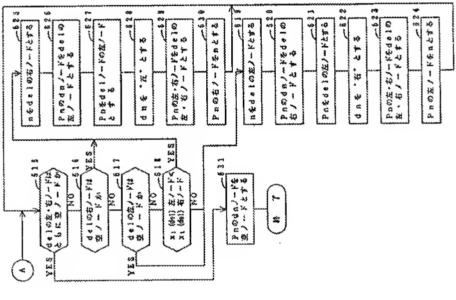
第5図

第4図

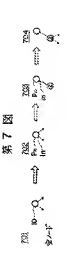
第 6 図



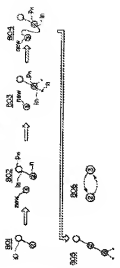
第 6 図



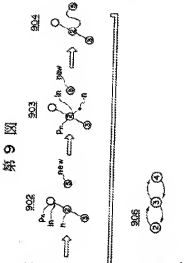
第7図



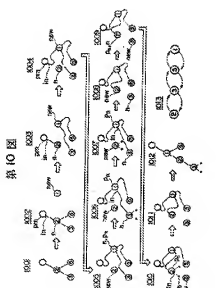
第8図



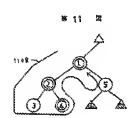
第9図



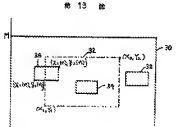
第10図



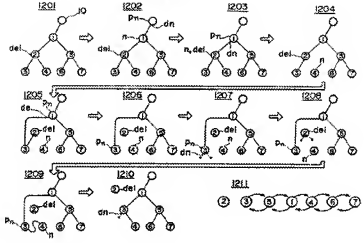
第11図



第13図



第 12 図


1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
del
dn
n